Special Edition

*The San Francisco Apple Core*

# *Cider Press* Gazette

Vol. 1 - - No. 1

## ON LINE!

S.F.A.C. Bulletin Board.

*The San Francisco Apple Core* now has an "official" electronic bulletin board.

Operated by Draco-Net, the board operates 24 hrs./day and is part of a larger system of commercial boards, all accessed through the same number.

(415) 474-8608

**Log-On:** The log-on procedure is simple and requires only that you answer the questions asked on the screen when you call.

**Access:** The board is a commercial board and charges a fee for access to its information. The fee, however, will be waived for members of the *San Francisco Apple Core.*

**Password:** You will be asked to assign yourself a password. In order to be recognized as a member of the S.F.A.C. you must make the last character of your password the "@"

You will be given official access after your membership in the S.F.A.C. has been verified.

**Note:** Ample help is available from the board itself. Printing a list of the commands is recommended.

The **Macintosh Interest Group** has been meeting two times each month, once after the general meeting and again in the east bay.

The next east bay meeting will be held on the 22nd of October at the Concord library, on the corner of Parkside and Salvio. Meeting time is 7:00 P.M.

A representative from **Telos** software will be present to demonstrate ***FileVision*** , one of the most innovative pieces of software for the Macintosh which makes use of the graphics capabilities and mouse interface.

***FileVision*** uses pictorial symbols of the users' own design to represent the subject, structure and organization of a collection of information. Using the mouse one can "point" to an image and either link to another level of pictorial representations or access the data at the current level.

Using pictures to represent topics and/or subject areas provides an unique way of classifying and cataloguing information. Creative uses for ***FileVision*** include filing anatomy information; parts-house warehouse information, and using the linking feature to create stories and adventure games.

Public domain software for the Macintosh will also be available at the meeting. For further information call the *S.F.A.C.* clubhouse at 566-2342 or Stan Guidero at 432-9713.

# B*A*S*I*C

Copyright (c) 1984, by Joyce Conklin

**FALL FUN**

We've been working on BASIC for quite a while now - I figured that it was time to show you some other things. So the first topic for study will be sorting and the Apple screen organization. Wait - wait - where did everybody go???!!!

O.K., since there are just the two of us, let's have a nice little chat about the weird things that Apple did with the screen as far as keeping track of screen locations in memory.

There is nothing weird about picking memory location 1024 for the first spot in the upper left corner of the screen; 1023 is the last number of the first K of memory (1023 equals 2 raised to the 10th power, right? and we work with the binary number system if we are silicon chips, right?).

The first K of memory is called zero page and is where the Apple is working when you are doing your programming - at least if you are only doing the little programs which have been listed in this column.

I could get sidetracked with a discussion and comparison of the Apple ][ reference manual (1979) and the DOS Programmer's manual (1982), but I'll save that for another time. At any rate, the upper left screen location is 1024 so the upper right screen location will be 1024 + 39 or 1063 - in the 40-column mode.

Ordinary people would think that the left-most position of the second row on the screen would be 1064 - but not Apple people! The text screen, for memory purposes, is divided into three sections; the first row in the second section is ordinary people's row 9 and the first row in the third section is ordinary people's row 17. As far as memory is concerned, the first position in the second row of the screen follows the last position of the 17th row of the screen.

Well ... almost follows! The actual numbering is as follows (extracted from *The Apple Users' Encyclopedia* by Gary Phillips, Joyce Conklin and Don Scellato.)

| Text Screen Line # | Decimal Address |
|---|---|
| 1 | 1024-1063 |
| 2 | 1152-1191 |
| 3 | 1280-1319 |
| 4 | 1408-1447 |
| 5 | 1536-1575 |
| 6 | 1664-1703 |
| 7 | 1792-1831 |
| 8 | 1920-1959 |
| 9 | 1064-1103 |
| 10 | 1192-1231 |
| 11 | 1320-1359 |
| 12 | 1448-1487 |
| 13 | 1576-1615 |
| 14 | 1704-1743 |
| 15 | 1832-1871 |
| 16 | 1960-1999 |
| 17 | 1104-1143 |
| 18 | 1232-1271 |
| 19 | 1360-1399 |
| 20 | 1488-1537 |
| 21 | 1616-1655 |
| 22 | 1744-1783 |
| 23 | 1872-1911 |
| 24 | 2000-2039 |

Notice that line 9 follows immediately after the locations in line 1, and line 17 follows immediately after those in line 9, but...! Line 17 ends with memory location 1143 and line 2 starts with memory location 1152 - that's eight memory locations unaccounted for. Got that?

Now for the $64,000 question (Trivial Pursuits question: what was Joyce Brother's topic when she was a contestant on that show?) -- WHY does Apple do it that way?? My reference sources do not say why, just what.

One suggestion was that it matches the arrangement of the individual pixels which make up a character on the 40-column screen (the grid is 8 high by 5 wide).

You can experiment with the screen locations by trying some POKE commands and see where your character shows up; e.g. POKE 1458,65 will place a capital A in the

11th position of the 12th row (equivalent to the BASIC line of VTAB 12: HTAB 11: PRINT "A").

This is preliminary to sharing a program with you which will demonstrate sorting on the screen - you can see the characters with lower ASCII values "bubble up" from the end of the screen to the front, but they do follow the screen memory locations so you will see the alphabet split along the lines discussed above.

There is one more thing which you must do before running the BASIC program, and that is to load the actual sort routine which is a machine language routine stored at memory location $8000 (32768). Wait!! There are only the two of us, and I promise not to laugh if you cry about machine language and THE monitor and stuff like that - I did the same thing when I first got my Apple too. We'll take it slowly and it won't be painful at all.

As mentioned in the first paragraph, silicon chips only speak binary or base-2 language. That is because the computer only understands electricity, which comes in two forms (+ and -, or On and Off -- doesn't matter what you call them.

> [Or even if you mess it up and call the electrons positive which is esentially what Ben Franklin did when he thought electricity was a fluid, thus confusing the heck out of generations of chemists when they encounter electronics for the first time. But that's another story!]).

The computer speaks binary very well, but it can't load itself (Hal and 2001 notwithstanding); humans don't speak binary very well at all. The compromise is hexadecimal arithmetic and Assembly language, and that's where THE monitor comes in.

This refers to a set of built-in or ROM routines or programs which control how the computer operates, not to the screen on which you see a copy (sort-of) of what is going on inside the computer. Assembly language consists of some 50+ commands, each of which is represented by a hexadecimal "token"; this information can be "translated" into binary by the computer and then the program will run as fast as the electricity can travel instead of te relative snail's pace of BASIC.

The trade-off is that we falible carbon-based systems can understand BASIC! (At least if you've been reading this column for any length of time, you should be able to.)

You can shift from Applesoft (the ] prompt) to the monitor ( * prompt) by typing CALL -151. This causes the computer to execute the built-in routine at location -151 ($FF69 or 65385). Then you can type in the following information exactly - repeat, exactly - as you see it below. Be sure to put a colon after the address, NO space between the colon and the first hex number, a space between each of the hexadecimal values, and a <Return> at the end of each of the lines.

When you have finished typing in the hex values, and have checked the screen for typos, type 3D0G (that's a zero, not the letter O) and you ill once more have the Applesoft prompt of ]. (The hard part, if you don't have an Assembler program, is that if you find a mistake you will have to retype the entire line where the mistake occurred.

If it is an ommission, you will have to retype from the beginning of the error line TO THE END of the program.) NOW -- type BSAVE MACHSORT, A$8000, L$50 <RETURN>. This will save your sort routine to disk. You will have to BRUN MACHSORT before the demonstration below will work.

```
8000:A9 00 85 01 A9 04 85 02

8008:A9 D0 85 03 A9 07 85 04

8010:A0 00 B1 01 85 05 C8 B1

8018:01 18 C5 05 90 14 18 A5

8020:01 69 01 85 01 A5 02 69

8028:00 85 02 C5 04 F0 10 4C

8030:10 80 88 91 01 C8 A5 05

8038:91 10 A2 FF 4C 1E 80 A5

8040:01 C5 03 D0 CB E8 F0 B8

8048:60 00 00 00 00 00 00 00
```

Take a deep breath, congratulate yourself on your first machine language program, and get ready to see what is happening when you sort something. First, BRUN MACHSORT to put the sort routine in memory. Then RUN SORTDEMO.

You will see the random assortment of ASCII characters appear on the screen, one location at a time and you can see the every-eight-line pattern we talked about. Then the sort routine is activated and you can see the characters being arranged by ASCII code from 01 (inverse A) to 255 (lowercase z).

```
100  REM  PROGRAM WRITTEN BY
     EMMANUEL HUNA
```

```
110  REM  IF YOU WISH TO TAKE A LOOK AT
     THE MACHINE LANGUAGE SORT, TYPE IN
     CALL -151, AND THEN 8000L.
     For a double check that you have
     typed in the machine language pro-
     gram correctly, see the assembled
     version below.

120  FOR X = 1024 TO 2000
     Lines 120-150 fill the screen with
     random ASCII characters

130 A =  RND (1) * 255

140  POKE X,A

150  NEXT X

160  CALL  - 32768

170  GET A$: HOME
     Press any key and start the whole
     thing all over again!

180  GOTO 120
```

If you have loaded (well, BLOADed) the machine language sort routine, you can inspect what is actually in memory by typing CALL -151 (to get into the monitor) and then typing 8000L <Return>. You will see one screen of information, showing the memory address at the left, the hexadecimal tokens in the next two to eight columns and the Assembly language mnemonics on the right.

```
8000- A9 00     LDA  #$00
8002- 85 01     STA  $01
8004- A9 04     LDA  #$04
8006- 85 02     STA  $02
8008- A9 D0     LDA  #$D0
800A- 85 03     STA  $03
800C- A9 07     LDA  #$07
800E- 85 04     STA  $04
8010- A0 00     LDY  #$00
8012- B1 01     LDA  ($01),Y
8014- 84 05     STA  $05
8016- C8        INY
8017- B1 01     LDA  ($01),Y
8019- 18        CLC
801A- C5 05     CMP  $05
801C- 90 14     BCC  $8032
801E- 18        CLC
801F- A5 01     LDA  $01
8021- 69 01     ADC  #$01
8023- 85 01     STA  $01
```

This is all that will appear on one screen; you will have the * prompt waiting at the bottom. To see the next screen of assembled listing, just type L <Return>.

```
8025- A5 02     LDA  $02
8027- 69 00     ADC  #$00
8029- 85 02     STA  $02
802B- C5 04     CMP  $04
802D- F0 10     BEQ  $803F
802F- 4C 10 80  JMP  $8010
8032- 80        DEY
8033- 91 01     STA  ($01),Y
8035- C8        INY
8036- A5 05     LDA  $05
8038- 91 01     STA  ($01),Y
803A- A2 FF     LDX  #$FF
803C- 4C 1E 80  JMP  $801E
803F- A5 01     LDA  $01
8041- C5 03     CMP  $03
8043- D0 CB     BNE  $8010
8045- E8        INX
8046- F0 B8     BEQ  $8000
8048- 60        RTS
8049- 00        BRK
```

The lines after this, if you type another L <Return>, will all be 00 and BRK, meaning nothing is stored there. If all this inner working of the Apple appeals to you, check out Roger Wagner's book "Assembly Lines" wherein there is a more complete (and certainly more learned) explanation, along with a bunch of neat tables, arranging op codes numerically and mnemonics alphabetically.

---

In the June column, I mentioned that I had a handy-dandy menu program which would RUN a program with a single key-stroke. I have finally found the program, hidden on one of my many unlabeled disks (well, not totally - there was a scrap of paper with the catalog stuffed in the envelope!).

```
20  POKE 1012,0

    1012 is the Autostart ROM Power up
    mask - does anyone know what this
    means??!!

30  TEXT : HOME

40  POKE 50,128

    50 sets the screen display mode; 255=
    normal, 127=flashing,63=inverse

50  INPUT A$

60  IF A$ = "QWERTY" THEN  NORMAL :
    GOTO 90

70  FOR I = 1 TO 10: READ T: POKE 767 +
    I,T: NEXT I: CALL 768

80  DATA  32,96,251,32,58,255,76,3,3,0
```

I was going to insert the usual REM comment here, but since I wasn't sure exactly what these two lines (70-80) did, I had to run it myself. Now I think I'll let you do the same, because it's fun to find out the result! A clue: 768 is the usual location for the sound-making subroutines.

```
90  TEXT : HOME :D$ =  CHR$ (4): PRINT
    D$"CATALOG":B =  PEEK (37) - 2: IF B
    > 22 THEN B = 22
```

PEEK 37 defines the current vertical position of the cursor

```
100 T = 0:CH = 4: FOR CV = 0 TO 23:
    GOSUB 230: IF C <  > 160 THEN  POKE
    P - 1,219: POKE P,T + 193: POKE
    P + 1,221:T = T + 1:S = CV
```

This plus the subroutine at 230 make the menu have the letters in [ ] brackets

```
110  NEXT CV: VTAB 24:A$ = "TYPE LETTER
     TO RUN, OR LOAD=1 LOCK=2 UNLOCK=3
     DELETE=4 EXIT=5..."

120 B$ = "RUN": HTAB 1: PRINT  LEFT$
     (A$,39);:A$ =  MID$ (A$,2) +  LEFT$
     (A$,1):K =  PEEK ( - 16384): IF K <
     128 THEN  FOR K = 1 TO 75: NEXT K:K
     =  FRE (0): GOTO 120

130  POKE  - 16368,0:K = K - 176: IF K <
     1 OR K > 5 THEN 200
```

-16368,0 is our old friend from previous columns, the value to clear the keyboard strobe

```
140  HTAB 1: CALL  - 868: IF K = 5 THEN
     END
```

CALL -868 clears to the end of the current line

```
150  PRINT " PRESS 'LETTER' YOU WISH TO
     ";: IF K = 1 THEN B$ = "LOAD"

160  IF K = 2 THEN B$ = "LOCK"

170  IF K = 3 THEN B$ = "UNLOCK"

180  IF K = 4 THEN B$ = "DELETE": FLASH

190  PRINT B$;: CALL  - 198: NORMAL :
     GET K$:K =  ASC (K$) - 48
```

CALL -198 rings the bell

```
200  IF K < 17 OR K > T + 16 THEN 120

210 CH = 1:CV = S - T + K - 16: GOSUB
     230:  IF  C  = 194 AND (B$ = "RUN" OR
B$
     = "LOAD") THEN B$ = "B" + B$

220  FOR CH = 6 TO 39: GOSUB 230:B$ = B$
     +  CHR$ (C): NEXT CH: HTAB 1: CALL
     - 868: PRINT B$: PRINT D$;B$:
     GOTO 90

230 C1 =  INT (CV / 8):C2 = CV - C1 * 8:
     P = 1024 + 128 * C2 + 40 * C1 + CH:
     C =  PEEK (P): RETURN
```

---

In case you have never seen all the characters in the primary ASCII set, run the following program to see them displayed in order.

```
10 TEXT: HOME
20  FOR L = 0 TO 255: PRINT CHR$(L);: NEXT
L: PRINT
30 END
```

Oooops! What is that all about? You only had one loop from 0 through 255 and you got two listings of essentially the same thing. Does Apple have something (else!) weird going on?

Try this one:

```
10 TEXT: HOME
20 FOR L = 0 TO 255
30 POKE 1024 + L, L
40 NEXT
50 VTAB 23: END
```

Now you have the inverse and the normal characters, but the flashing characters are missing and the letters following lower-case w are dropped into the previously mentioned missing eight memory locations. AND you don't hear the bell or see a blank line, indicating that the control characters are being "printed."

Here's the problem for the month (or however long it takes for me to find or receive and answer!): What is the total explanation for what is going on? Why no flashing characters in this program the way there were in the longer one which was part of the sort demo? Techies, arise (or to the rescue, whichever!)


**LISTEN to**
***"The Voice of Experience"***

# SynPhonix™

**Electronic Speech Articulator for the Apple!**

***Hear at last,*** hardware synthesis at software prices. SynPhonix™ produces unparalleled voice quality using the ***all new SSI263*** phoneme synthesizer with 64 phonemes, 100's of allophones, plus software controlled pitch, amplitude, rate and filter. The SONIX™ 65© Speech Operating System puts you in control of it all. SynPhonix™ is brought to you by professionals with more than a decade of experience in speech synthesis products. Applications are abundant. SynPhonix™ features include:

- SSI263 Phoneme Synthesizer
- Unlimited vocabulary
- High Quality Phonetic Voice
- Low Data Rate
- Singing/Sound Effects
- Comprehensive User's Manual
- Demo Software Included
- Free Spelling Program
- Transparent to User's Software
- SONIX™ 65©
  - Speech Operating System
  - Speech Editor
  - Preprogrammed Vocabulary
  - Vocabulary File Manager
  - EASY TO USE
- 90 Day Limited Warranty
- Suggested Retail $125.00

*Dealer inquiries welcome!*

**Artic Technologies**
*"The Voice of Experience"*

2234 Star Ct.-2D (313) 852-8344
Auburn Heights, Michigan 48057

Apple is a Trademark of Apple Computer, Inc.

The East Bay group has been re-organized with a new location in Berkeley at the Winner's Circle Systems Inc. 2816 Telegraph Ave., Berkeley.

Thanks to Andy Jung for providing a location.

Harold Brown is the new group leader and he and Andy have asked that anyone who is interested in attending the group meetings please call before coming to R.S.V.P. (428-1176)

The group will meet on the 1st Tuesday of each month with
the first meeting November 6th from 7:00 to 9:00

GREEN APPLES

Green Apples, an eight session course on Applesoft Basic is scheduled to begin on Thursday October 4th. Scheduled dates include October 4, October 25, November 1, November 29, December 6, December 27, January 3 and January 24. All classes are at the clubhouse 1515 Sloat Blvd. Suite 2 at 8:00 p.m.

Instructor, Keith Thomas, is starting the course from the very beginning. Those participants with some background in BASIC will have the opportunity to take advantage of a refresher course the first few meetings while new BASIC programmers establish a good sound foundation of the language.

There is no fee for the course. Registration will take place at the first meeting. If more information is required contact Keith Thomas 837-3164.

This course will be repeated in the Spring.

INTRODUCING *** PRO DOS ***

Ron Dell'Aquila is focusing his attentions these days toward PRO DOS. He is not abandoning his CP/M Interest Group, on the contrary....expanding it and welcomes all interested members to meet on the 4th Wednesday of each month.

PEEKS POKES AND CALLS - WHAT ARE THEY?

Monday November 19, 1984
S.F.A.C. Clubhouse
1515 Sloat Blvd. Suite 2
7:30 p.m.

Registration is required.
Register by mail or at the General Meeting.
A $5.00 refundable fee must accompany registration for each seminar.
Send name, address, telephone number and appropriate fees to:

S.F.A.C - SEMINAR
1515 Sloat Blvd. Suite 2
San Francisco, CA 94132

Make checks payable to San Francisco Apple Core.
Checks will be held until the Seminar and returned to the registrant upon check-in.

THE LIBRARY - REVISITED

A wealth of information and programs is available to members through the S.F.A.C. and I.A.C. libraries. An Interest Group is forming that will review the programs, write reviews and catalog documentation for the resource library.

This group will meet regularly on the third Thursday of each month, 7:30 p.m. at the Clubhouse. For more information contact Kathy Rudovsky at 756-6480.

GRAPHICS GRAPHICS GRAPHICS

A six to eight week course on computer graphics is now being offered to club members. The Graphics course is scheduled for the 2nd and 4th Mondays of the month. All meetings will be held at the Clubhouse, 1515 Sloat Blvd. Suite 2, at 7:30 p.m.

Questions regarding registration and course outline can be directed to the Graphics Instructor, Reed Harris, at 924-7221.

| | | | | |
|---|---|---|---|---|
| Lisa | Larry Clopp<br>333-3707 | Call for location | 1st & 3rd Wednesday | 6:00 pm |
| Marin CP/M | Jim Ayres<br>383-0473<br>383-5759 | 301 Poplar<br>Mill Valley | 2nd & 4th Tuesday | 7:30 pm |
| PRO DOS-CP/M | Ron Dell'Aquila<br>566-8460 | Clubhouse | 4th Wednesday | 7:30 pm |
| Dot Matrix Printers | Michael Fischer<br>(w) 557-1250<br>(h) 933-4435 | Clubhouse | 4th Tuesday | 7:30 pm |
| Forth | R. Dudley Ackerman<br>626-6295 | Clubhouse | 1st & 3rd Tuesday | 7:30 pm |
| Pascal | Mike Schnapp<br>386-2238 | Call for location | 4th Monday | 7:30 pm |
| Nuer's | Kathy Rudovsky<br>756-6480 | Clubhouse | 3rd Wednesday | 7:30 pm |
| Spreadsheets | Don Scellato<br>672-2051 | Clubhouse | 2nd Tuesday | 7:30 pm |
| East Bay Spreadsheets | Don Scellato<br>672-2051 | Softwaire Centre - Willows<br>Pleasant Hill | | 7:00 pm |
| Green Apples | Keith Thomas<br>837-3164 | Clubhouse | 1st 4th & 5th Thursday | 8:00 pm |
| East Bay | Harold Brown<br>428-1176 | Call for location | 1st Tuesday | 7:00 pm |
| Macintosh | Don Scellato<br>Alan Smerican<br>672-2051 | Fort Mason Bldg. C | 3rd Saturday | 12:00 pm |
| Product Review | Angela Allberry<br>391-2715 | Clubhouse | 2nd Thursday | 7:30 pm |
| Telecommunications | George Thornally<br>398-7926<br>776-2603 | Call for location | 2nd Tuesday | 7:30 pm |
| Graphics | Reed Harris<br>924-7221 | Clubhouse | | 7:30 pm |
| Library-Revisited | Kathy Rudovsky<br>756-6480 | Clubhouse | 3rd Thursday | 7:30 pm |
| East Bay 32 Bit Affiliate | Stan Guidero<br>432-9713 | Brueners-Buskirk<br>Pleasant Hill | 3rd Monday | 7:00 pm |

San Francisco Apple Core Clubhouse
1515 Sloat Blvd. Suite 2
San Francisco, CA 94132
(415) 566-2342

SF APPLE CORE
1515 Sloat Blvd., Suite 2
San Francisco, CA. 94132

# TIME VALUE

Bulk Rate
U.S. Postage
PAID
Permit #85
EL Cerrito, CA 94530

TO:

[redacted]

**LOW PRICE DISK DRIVES !!!!**

**DISTAR 5-1/4 INCH HALF HIGH DRIVES**

DIRECT DRIVE MOTOR.....

AUTOMATIC DOS SELECTION.....

POSITIVE LOCK LEVER.....

6 MONTH WARRANTY.....

**$160 + TAX**

PURCHASE AT MEETING OR ORDER FROM THE SFAC BY MAIL.....

**LOW PRICE DISK DRIVES !!!!**

**AMT 5-1/4 INCH HALF HIGH DRIVES**

DIRECT DRIVE MOTOR.....

PANASONIC MECHANISM.....

WARRANTY.....

**$189 + TAX**

PURCHASE AT MEETING OR ORDER FROM THE SFAC BY MAIL.....

## For Sale

One Apple II Plus 48k, W/manual & AC cord
One Apple Language Card 16K, W/Manual
One M&R 80 Column Card, W/Manual & Disk
One AppleWriter II Disk, Master & Back-up, W/Manual

Quality assurance report 7/17/84 from Quest Computers, Millbrae.
This system has always been used on an ISO-2 (isolator)
and fan cooled.
**Price: $450.00 Call Brian (415) 359-5526 (6:00-9:00 PM)**